**¿QUÉ ES UNA COMPUTADORA?**

Dispositivo electrónico utilizado para procesar información y obtener resultados

DATOS DE ENTRADA 🡪 PROGRAMAS 🡪 DATOS DE SALIDA

**LA MEMORIA CENTRAL**

La información almacenada en memoria puede ser de dos tipos: las instrucciones de un programa y los datos con los que operan las instrucciones.

Para que un programa se pueda ejecutar, debe ser situado en la memoria central, en una operación denominada carga (load) del programa.

Cuando se ejecuta el programa, cualquier dato a procesar por el programa debe llevarse a memoria mediante instrucciones del programa.

La unidad central de memoria se llama byte (octeto). Un byte tiene la capacidad de almacenar un carácter de información, y esta formada por 8 bits, es decir, dígitos binarios.

Existen dos conceptos importantes asociados a cada celda o posición de memoria: su dirección y su contenido. Cada celda o byte tiene asociada una única dirección que indica su posición relativa en memoria mediante la cual se puede acceder a la posición para almacenar o recuperar información. La información almacenada en una posición de memoria es su contenido.

Siempre que una nueva información se almacena en una posición, se destruye (desaparece) cualquier información que en ella hubiera y no se puede recuperar.

La dirección es permanente y única, el contenido puede cambiar mientras se ejecuta un programa.

En la memoria principal se almacenan:

* Los datos enviados para procesarse desde los dispositivos de entrada
* Los programas que realizan procesos
* Los resultados obtenidos preparados para enviarse a un dispositivo de salida

Cuando un programa se ejecuta, se debe situar primero en memoria central de igual modo que los datos. Sin embargo, la información almacenada en la memoria se pierde cuando se apaga la PC y, por otra parte, la memoria central es limitada en capacidad. Por esta razón, para disponer de almacenamientos permanente, tanto para programas como para datos, se necesitan dispositivos auxiliares.

La memoria central es mucho más rápida y cara que la auxiliar. Se deben transferir los datos desde la memoria central antes de que puedan ser procesados. Los datos en memoria central son volátiles y desaparecen cuando se apaga la PC. Los datos de memoria auxiliar son permanentes y no desaparecen cuando la PC se apaga.

Dispositivos de entrada

Memoria central

Unidad de disco

Dispositivos de salida

**CONCEPTO DE ALGORITMO**

El programador de computadora es antes que nada una persona que resuelve problemas.

La metodología necesaria para resolver problemas mediante programas, concepto que se denomina metodología de programación

UN ALGORITMO ES UN METODO PARA RESOLVER UN PROBLEMA

PROBLEMA 🡪 DISEÑO DEL ALGORITMO 🡪 PROGRAMA DE COMPUTADORA

Los pasos para la resolución de un problema son:

1. DISEÑO DE ALGORITMO que describe una secuencia ordenada de pasos (sin ambigüedades) que conducen a la solución de un problema dado (análisis del problema y desarrollo del algoritmo)
2. Expresar el algoritmo como un programa en un lenguaje de programación adecuado (fase de codificación)
3. Ejecución y validación del programa por la computadora

Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir programa.

Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta.

Por ejemplo, una receta de cocina se puede escribir en cualquier idioma, los pasos son los mismos y el resultado es idéntico.

Un lenguaje de programación es tan solo un medio para expresar un algoritmo y una computadora es solo un procesador para ejecutarlo.

**CARACTERISTICAS DE LOS ALGORITMOS**

* Un algoritmo debe ser preciso a indicar el orden de realización de cada paso
* Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
* Un algoritmo debe ser finito. Si se sigue el mismo algoritmo se debe terminar en algún momento; o sea debe tener un numero finito de pasos.

ENTRADA 🡪 PROCESO 🡪 SALIDA

**LENGUAJES DE PROGRAMACION**

El procesador deber ser capa de INTERPRETAR el algoritmo, lo que significa:

* Comprender las instrucciones de cada paso
* Realizar las operaciones correspondientes

El algoritmo se ha de expresar en un formato que se denomina PROGRAMA.

Los principales tipos de lenguajes utilizados son:

* Lenguaje maquina
* Lenguaje de bajo nivel (ensamblador)
* Lenguaje de alto nivel

Los diferentes pasos (acciones) de un algoritmo se expresan en los programas como instrucciones, sentencias o proposiciones.

Un programa consta de una secuencia de instrucciones, cada una de las cuales especifica ciertas operaciones que debe ejecutar la computadora.

* Instrucciones de entrada y salida: instrucciones de transferencia de información y datos entre dispositivos periféricos y la memoria central
* Instrucciones aritmético - lógicas: instrucciones que ejecutan operaciones aritméticas y lógicas
* Instrucciones selectivas: instrucciones que permiten la selección de tareas alternativas en función de los resultados de diferentes expresiones condicionales
* Instrucciones repetitivas: instrucciones que permiten la repetición de secuencias de instrucciones un número determinado de veces

**LENGUAJES DE MAQUINA**

Están escritos en lenguajes directamente inteligibles por la maquina ya que sus instrucciones son cadenas binarias que especifican una operación, y las posiciones de memoria implicadas en la operación se denomina instrucciones de maquina o código máquina.

Las instrucciones en lenguaje máquina dependen del hardware de la computadora y por lo tanto, difieren de una computadora a otra.

Los inconvenientes superan las ventajas: no necesitan traducción, pero son difíciles y lentos para codificarse, son poco fiables y difíciles de verificar.

**LENGUAJES DE BAJO NIVEL**

También dependen de la maquina en particular. El lenguaje de bajo nivel por excelencia es el ENSAMBLADOR (*assembly languaje*). Las instrucciones en lenguaje ensamblador son instrucciones conocidas como nemotécnicas.

Una instrucción típica de suma seria:

ADD M, N, P

Su equivalente en código maquina es:

0110 1001 1010 1011

Un programa escrito en lenguaje ensamblador no puede ser ejecutado directamente por la computadora, sino que requiere una fase de traducción al lenguaje máquina.

El programa original escrito en lenguaje ensamblador se denomina programa fuente y el programa traducido en lenguaje maquina se conoce como programa objeto. El traductor de programas fuente a objeto se llama ENSAMBLADOR y es un programa existente en todas las computadoras.

Programa Objeto en Código Maquina

Programa Fuente en Ensamblador (ASSEMBLY)

Programa Ensamblador (ASSEMBLER)

Los lenguajes ensambladores poseen mayor facilidad de codificación, pero siguen teniendo dependencia total del lenguaje, además, la formación de los programas es más compleja que la correspondiente a los programadores de alto nivel, ya que exige técnicas de programación y conocimiento del interior de la máquina.

Hoy en día tienen aplicaciones de control de procesos, dispositivos electrónicos y aplicaciones de tiempo real.

**LENGUAJES DE ALTO NIVEL**

Son los más utilizados y están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina. Además, es independiente de la maquina por lo que son portables y transportables.

Ventajas:

* El tiempo de formación de los programadores es mas corto que en otros lenguajes
* La escritura se basa en reglas sintácticas similares a los lenguajes humanos
* Las modificaciones de los programas son más fáciles
* Reducción del corte de los programas
* Transportabilidad

Desventajas:

* Aumento del tiempo de puesta a punto, al necesitarse diferentes traducciones del programa fuente para conseguir el programa definitivo
* No se aprovechan los recursos internos de la maquina
* Aumento de la ocupación de la memoria
* El tiempo de ejecución de los programas es mucho mayor

Los programas fuente tienen que ser traducidos por los traductores, llamados compiladores y traductores o intérpretes.

Ejemplos de lenguajes de alto nivel: C, C++, COBOL, FORTRAN, PASCAL, Visual Basic, etc.

**TRADUCTORES DE LENGUAJE**

Son programas que traducen a su vez programas fuente escritos en lenguajes de alto nivel a código maquina

**INTERPRETES**

Es un traductor que toma un programa fuente, lo traduce y a continuación lo ejecuta. Los programas interpretes ya no se utilizan. No guardan resultado.

Equivale a un interprete humano que traduce frases que oye sobre la marcha, sin producir un escrito permanente.

PROGRAMA FUENTE 🡪 INTERPRETE 🡪 TRADUCCION Y EJECUCION LINEA A LINEA

Es más lento, puede usarse por ejemplo en PC’s y en la Play. Ejemplo, Basic, Java, Smalltalk.

**COMPILADORES**

Un compilador es un programa que traduce los programas fuente escritor en lenguaje de alto nivel a lenguaje máquina.

Los programas escritos en lenguaje de alto nivel se llaman PROGRAMAS FUENTE y el programa traducido PROGRAMA OBJETO o CODIGO OBJETO. El compilador traduce sentencia por sentencia el programa fuente.

Equivale a un traductor humano que toma un libro y produce otro equivalente escrito en otra lengua.

PROGRAMA FUENTE 🡪 COMPILADOR 🡪 PROGRAMA OBJETO

Ejemplo, C, C++, Pascal, Java y COBOL. (Java se pasa primero a byte code y luego se interpreta)

**LA COMPILACION Y SUS FASES**

El proceso de ejecución de un programa escrito en un lenguaje de programación y mediante un compilador suele tener los siguientes pasos:

1. Escritura del PROGRAMA FUENTE con un editor y guardarlo.
2. Introducir el programa fuente en memoria.
3. Compilar el programa en el compilador C.
4. Verificar y corregir errores de compilación.
5. Obtención del programa objeto.
6. El enlazador (Linker) obtiene el programa EJECUTABLE.
7. Se ejecuta el programa y, si no existen errores, se tendrá la salida del programa

PROGRAMA FUENTE

COMPILADOR (TRADUCTOR)

PROGRAMA OBJETO

MONTADOR

PROGRAMA EJECUTABLE EN LENGUAJE MAQUINA

En diagrama de flujo sería:

**EL LENGUAJE C**

C es un lenguaje de alto nivel de programación de propósito general. No está asociado a ningún sistema operativo, ni a ninguna maquina en especial.

C es una evolución de los lenguajes BCPL (Martin Richards) y B (Ken Thompson en 1970) para el primitivo Unix de la computadora DEC PDP-7

C nació realmente en 1978, con la publicación de *The C programming Lenguaje*, por Brian Kernighan y Dennis Ritchie (Patience Hall 1978)

En 1983, El American National Estandar Institute (ANSI) creo el comité con la tarea de hacer “una definición no ambigua del lenguaje C, e independiente de la maquina”, creando el estándar ANSI del lenguaje C

Se define como un lenguaje de programación estructurado con una enorme cantidad de potencia y flexibilidad.

**VENTAJAS DE C**

* Es poderoso y flexible, con órdenes, operaciones y funciones de biblioteca.
* Se pueden utilizar para desarrollar sistemas operativos, compiladores, sistemas en tiempo real y aplicaciones de comunicaciones
* Es portable y veloz para ejecutar
* Se han escrito numerosas librerías para C

Versiones de C 🡪 C ++ y C #

**CUESTIONARIO**

1. ¿Qué es una computadora?
2. ¿Dónde debe cargarse un programa para que se pueda ejecutar?
3. Enumere y explique los dos conceptos asociados a una celda de memoria
4. ¿Qué se guarda en la memoria principal?
5. ¿Qué es un algoritmo?
6. ¿Cuáles son los pasos para la resolución de un problema?
7. Enumere las características de un algoritmo
8. ¿Cuáles son los tipos de lenguajes?
9. ¿Cuáles son los tipos de instrucciones?
10. ¿Qué es un lenguaje de maquina?
11. ¿Qué es un lenguaje de bajo nivel?
12. ¿Qué es un lenguaje de alto nivel?
13. ¿Qué diferencia hay entre un intérprete y un compilador?
14. Enumere las fases de la compilación
15. Explique brevemente la historia del lenguaje C y sus ventajas.
16. ¿Qué es un ensamblador?
17. ¿Qué es un compilador?
18. ¿Qué es un intérprete?
19. Indique las diferencias entre cada uno de ellos